

video

video

~~String~~

→ message = ""

""" # "" is used  
in very long sentence

→ length of String

course = "Python"

→ course[0] → P

→ course[-1] → n

→ \ ⇒ escape character

→ course = "Python \ Programming" → Python & programming

→ \", \', \\", \n  
↓ ↓ ↓ ↓  
" , \ newline

→ course.upper() → PYTHON

→ course.lower() → python

→ course.title() → Python

→ course = " python"

→ course.strip() → Python # remove whitespace (rstrip | lstrip)

→ course.find("o") → 5

→ replace. replace("P", "j") → jython

→ print("p" in course) → True

→ print("g" not in course) → True

## Numbers

- $x = 1 + 2j$  #  $a + bi$
- augmented operator  $+=, -=, *=, /=$
- $\text{round}(2.9) = 3$
- $\text{abs}(-2.9) = 2.9$
- `import math`  
 $\text{math.ceil}(2.2) = 3$
- type conversion :-

`int(x), float(x), bool(x), str(x)`

→ `type(x) → <class 'str'>`

→ Falsy value  $\Rightarrow$  "", 0, None

→ `bool(0) ⇒ False`    `bool(5) ⇒ True`

→ Ternary operator

`message = "Eligible" if age >= 18 else "Not eligible"`

## §. Type of function.

`message = "work"`

`file = open("content.txt", "w")`

`file.write(message)`

~~default parameter~~  
de



## default parameter

```
def increment(number, by=1):  
    return number + by
```

if we print

`print(increment(2))` → 3

`printor(increment(2, 5))` → 7

→ all default parameter is write in end of the main parameter.

args

```
def multiply(*numbers):
```

```
    total = 1
```

```
    for number in numbers:
```

```
        total *= number
```

```
    return total
```

```
print(multiply(2, 3, 4, 5))
```

~~print~~